

A Performance Comparison Between Block Interleaved and Helically Interleaved Concatenated Coding Systems

K.-M. Cheung and L. Swanson
Communications Systems Research Section

This article compares the performance (bit-error rate vs. signal-to-noise ratio) of two different interleaving systems, block interleaving and the newer helical interleaving. Both systems are studied with and without error forecasting. Without error forecasting, the two systems have identical performance. When error forecasting is used with shallow interleaving, helical interleaving gains, but less than 0.05 dB, over block interleaving. For higher interleaving depth, the systems have almost indistinguishable performance.

I. Introduction

As NASA wants to receive more data from planetary missions, and as technologies like data compression make projects tighten their error-rate requirements, many missions are turning to concatenated coding schemes with Reed-Solomon (RS) codes as "outer" codes concatenated with "inner" convolutional codes, which have been used in the Deep Space Network for years (Fig. 1). RS decoders for use beginning in the mid-1990s will be installed in the network by February 1, 1992.

When concatenated coding is used, symbols in RS words are generally "interleaved," so that the error cor-

recting ability of one RS word is not used up entirely by one or two long error bursts from the Viterbi (convolutional) decoder. Some missions planning to use RS codes in the Deep Space Network will have "block" interleaving depth five. This means that consecutive symbols in a RS word are separated on the channel by exactly four other RS symbols, one from each of the other four words in the block (Fig. 2).

Recently, other interleaving schemes have been suggested, such as E. Berlekamp's "helical interleaving" [1, 2]. In this article, the initial analysis of this scheme is presented. In helical interleaving, words go into the interleaver/deinterleaver in the staggered way shown in Fig 3.

(Notice that if a block-interleaved code block were pasted onto a cylinder, the set of symbols consisting of the j th symbol in each word lies on a circle around the cylinder; if the helically interleaved words were pasted onto a cylinder, the set consisting of the j th symbol in each word lies on a helix.) Helical interleaving is defined precisely in Section II.

It is worth noticing that for a given interleaving depth, the two schemes yield statistically identical code performance. As far as any one codeword is concerned, it consists of certain RS symbols; its decoding does not depend on which other codewords the intervening symbols belong to. So why this apparently more complicated interleaving scheme? There are two ways in which we might expect helical interleaving to be different from block interleaving. The first is synchronization: if a synchronization marker is placed before or at the beginning of each codeword in a block-interleaved block, then several markers appear consecutively at the beginning of each code block; with helical interleaving, the markers would be uniformly scattered. This should allow faster (though possibly more complicated) synchronization, but we do not address this issue here. The second way in which helical interleaving can be expected to perform differently from block interleaving is in error forecasting. For several years, the DSN's planning for RS decoders has taken into account the slight performance gain available from the fact that RS codes can use quality information in decoding; a symbol that is believed questionable and flagged as an "erasure" costs only half an error [3].

The error-forecasting scheme used in this article is a simple but effective one: that is to create erasures in those words that have too many symbol errors [4] by assuming that errors in adjacent words continue in the undecodable word. Based on the above idea, there are two ways to flag an erasure; that is, either flag a symbol of an undecodable word as an erasure if either one of the code symbols next to it on the channel is detected as erroneous (single-sided forecasting), or flag a symbol as an erasure only if the code symbols on both sides of it on the channel are detected as erroneous (double-sided forecasting). Simulations show that in all cases the more aggressive single-sided forecasting scheme performs better than the conservative double-sided forecasting scheme. Thus we choose to use the single-sided forecasting scheme in all our simulations in this article. Since the Viterbi decoder, which decodes the inner convolutional code, creates errors in bursts, previous analysis [5] shows that in the case of block interleaving, using this simple error-forecasting strategy allows a gain of about 0.04 dB.

For block interleaved data, words go into the interleaver as shown in Fig. 2. Each interleaving block is separated from others, and all possible error forecasting can be done within the same block. Thus a finite buffer can do all error forecasting. In the light of this finite buffer feature, error forecasting with redecoding to each interleaving block was selected, disregarding the order of decoding of codewords in each block. That is, when a codeword in a block is decoded successfully, erasure information is generated for both of its adjacent words in the block. After the first round of decoding in a block, the previously undecoded codewords are then redecoded using the additional erasure information. This process goes on until no more undecodable words can be recovered. For helically interleaved data, words go into the interleaver in the staggered way shown in Fig. 3. Unlike the block interleaving scheme in which each codeword is "related" only to words in the same block, each codeword in the helical interleaving scheme is theoretically "related" to all words before and after it, and this makes error forecasting with redecoding impractical, if not impossible, since it would require a buffer with infinite memory. Thus in helical interleaving one does error forecasting without redecoding. That is, when a codeword is successfully decoded, erasure information is generated to those symbols (of other not yet decoded codewords) that are adjacent to the previously erroneous symbols of the decoded codeword. Each codeword is then decoded in order, first without using the erasure information. If the codeword is decodable, the decoder goes on to decode the next codeword. If not, the codeword is decoded once again using the erasure information generated from previously decoded words. Whether the codeword is decodable or not (with erasure information), the decoder goes on to decode the next codeword.

For deep interleaving, both block and helical interleaving give nearly identical error forecasting capability because both schemes are essentially "ideal interleaving" schemes which allow nearly statistically independent RS symbols. But for shallow helical interleaving (e.g., depth 2, which is used by Galileo), a word with too many symbol errors to decode is preceded by a part of one word and part of another. On the one hand, this means that a Viterbi decoder error burst is less likely to keep both different codewords from decoding, and so the error forecasting capability should be enhanced. On the other hand, only some of the symbols in a given codeword are preceded by symbols in a codeword that is decoded before it. For the exact schemes we compare in Section III, the trade-off between these effects depends on the interleaving depths and the inner convolutional codes. For interleaving depth 2, helical interleaving gives a slight performance gain over block interleaving with both the $(7,1/2)$ and $(15,1/4)$ inner

convolutional codes. For interleaving depth 4 or 8, helical interleaving is slightly better or slightly worse than block interleaving, depending on whether the (15,1/4) code or the (7,1/2) code is used as the inner code.

II. Helical Interleaving

In this section, mathematical definitions of the interleaving schemes are included for completeness, but the reader is advised to consider Figs. 2 and 3 to illustrate these definitions.

Definition 1. A block code with n code symbols per codeword is block interleaved to depth d if the code symbols sent on the channel are packed into codewords in such a way that the code symbol that follows (on the channel) the j th symbol of the m th codeword, $m < d$, is the j th code symbol of the $(m+1)$ th codeword, and the code symbol that follows the j th code symbol of the d th codeword, $j < n$, is the $(j+1)$ th code symbol of the first codeword, and the code symbol following the n th code symbol of the d th codeword is the first symbol of the first codeword in the next block.

Definition 2. A block code with n code symbols per codeword is helically interleaved to depth d , where $dr = n - 1$ and r is any integer, if code symbols are packed into codewords in such a way that the code symbol that follows (on the channel) the j th code symbol of the m th word, $m < d$, is the $(j-r)$ th code symbol of the $(m+1)$ th codeword; if $j-r \leq 0$, then this is the $(j-r+n)$ th code symbol of the m th codeword of the "previous" group. The j th code symbol of the d th codeword is followed by the $(j+(d-1)r)$ th code symbol of the first codeword of this group if $j \leq r$ and by the $(j-r)$ th code symbol of the first codeword of the next group if $j > r$.

Helical interleaving can be defined for codeword length n and interleaving depth d as long as n and d are relatively prime. While this more complicated definition for interleaving depths 4 and 8 was needed in the simulations, technical definitions will not be written here. The interested reader can find them in [1].

Helical interleaving was introduced in 1982 [1, 2] by E. Berlekamp. It has since been successfully implemented in several hardware decoders. Figure 3 depicts a helical interleaver for a code of length 4 interleaved to depth 3. As discussed in Section I, use of a helical interleaving scheme

in general does not enhance the gain performance of a concatenated coding system.

For many applications, it is better than block interleaving because of synchronization and interleaving delay. Helical interleaving facilitates synchronization in the following way: in block interleaving, the j th symbols of two adjacent codewords in a block are adjacent to each other in the channel; in helical interleaving, the j th symbols of two adjacent codewords are separated from each other in the channel by $n - 1$ symbols in a block-interleaved system. If a synchronization marker is placed before or at the beginning of each codeword in a block-interleaved block, several markers appear consecutively at the beginning of each code block in the channel symbol streams, and the receiver can acquire only synchronization modulo nd . With helical interleaving, the markers are uniformly scattered, and this allows synchronization modulo n . The difference in interleaving delay is that the end-to-end delay of a block interleaver is $2nd$, exclusive of the channel delay, whereas the delay of a helical interleaver is only nd .

III. Simulation

The performance of the concatenated coding systems described in this section were obtained by simulation. The inner codes used were the standard (7,1/2) convolutional code, which was simulated with a software Viterbi decoder, and Galileo's experimental (15,1/4) code, which was decoded using the Communications Systems Research Section's long constraint length "Little Viterbi Decoder" built for Advanced Systems. The outer codes used were the standard (255,223) RS code, which has been adopted for use by a number of present and future missions, and the proposed (1023,959) RS code for deep-space missions of the future.

Figures 4 through 11 give the SNR performance comparisons of various concatenated schemes using block interleaving and helical interleaving. Figures 4 and 5 show the performance curves of the concatenated coding schemes using the (7,1/2) convolutional code as inner code, and the (255,223) RS code as outer code, without and with error forecasting respectively. Similarly, Figs. 6 and 7 show the performance curves of the concatenated coding schemes with the (7,1/2) convolutional code and the (1023,959) RS code. Figures 8 and 9 show the performance curves of the concatenated schemes using the (15,1/4) convolutional code and the (255,223) RS code. And finally, Figs. 10 and 11 show the performance curves of the concatenated coding schemes using the (15,1/4) convolutional code and the (1023,959) RS code.

IV. Conclusion

It is observed from Figs. 4 through 11 that, as pointed out in Section I, when no error-forecasting strategy is used, both helical interleaving and block interleaving perform identically for all interleaving depths. When error forecasting is used, helical interleaving gains less than 0.05 dB at shallow interleaving. For deeper interleaving (e.g., depth 4 or depth 8), helical interleaving is slightly worse or slightly better than block interleaving, depending on whether the (7,1/2) convolutional code or the (15,1/4) convolutional

code is used as inner code. The difference in performance between helical and block interleaving for higher depths might be attributed to the fact that the (15,1/4) code produces longer bursts than the (7,1/2) code.

On the basis of these results, it is recommended that helical interleaving be considered for possible use only in particular situations, and not for general use. As usual, the real moral seems to be that one should allow greater interleaving depth, whatever interleaving scheme is used.

References

- [1] Cyclotomics Corporation, *Interleaved Coding for Bursty Channels*, Final Report on Phase I, Small Business Innovation Research, NSF Grant No. ECS-8260180, Berkeley, California, April 12, 1983.
- [2] E. R. Berlekamp and P. Tong, *Improved Interleavers for Digital Communications*, United States Patent Number 4559625 (granted December 17, 1985).
- [3] R. E. Blahut, *Theory and Practice of Error Control Codes*, Reading, Massachusetts: Addison Wesley, 1983.
- [4] R. J. McEliece and L. Swanson, "On the Decoder Error Probability for Reed-Solomon Codes," *IEEE Trans. Inform. Theory*, IT-32, pp. 701-703, September 1986.
- [5] G. H. Pitt, III, and L. Swanson, "Erasure Information for a Reed-Solomon Decoder," *TDA Progress Report 42-83*, Jet Propulsion Laboratory, Pasadena, California, pp. 39-44, November 15, 1985.

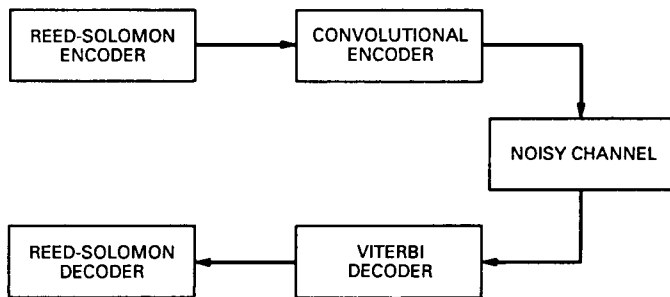


Fig. 1. Concatenated coding system.

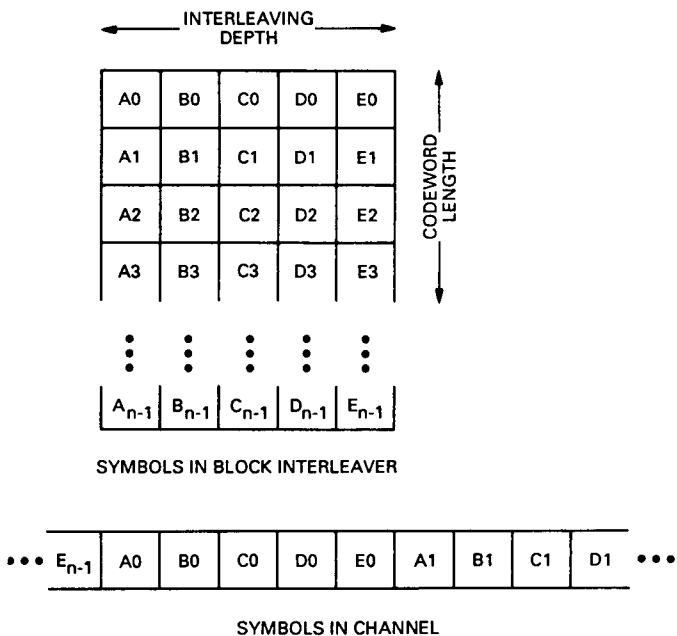


Fig. 2. Example of block interleaving.

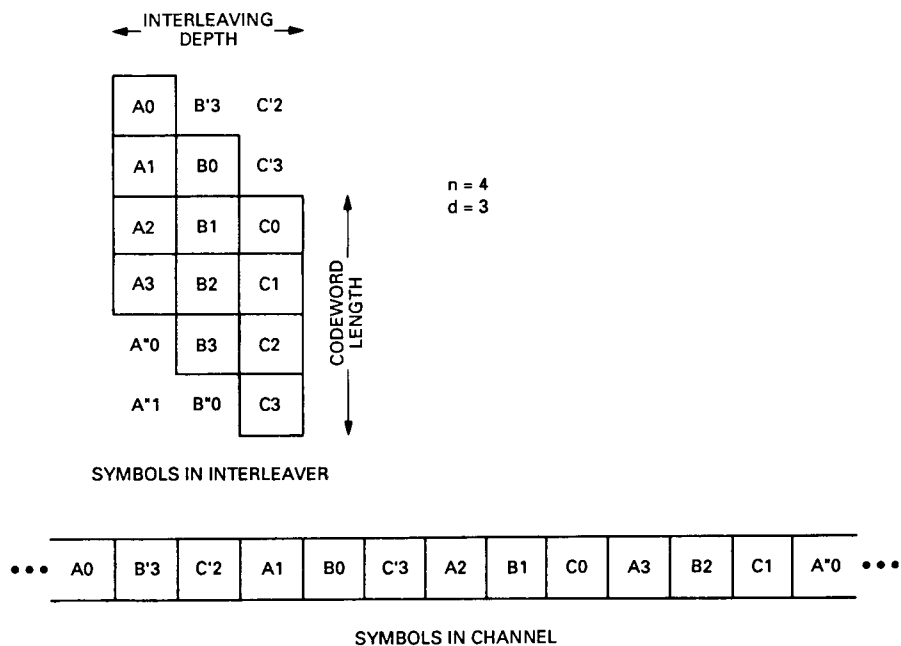


Fig. 3. Example of helical interleaving.

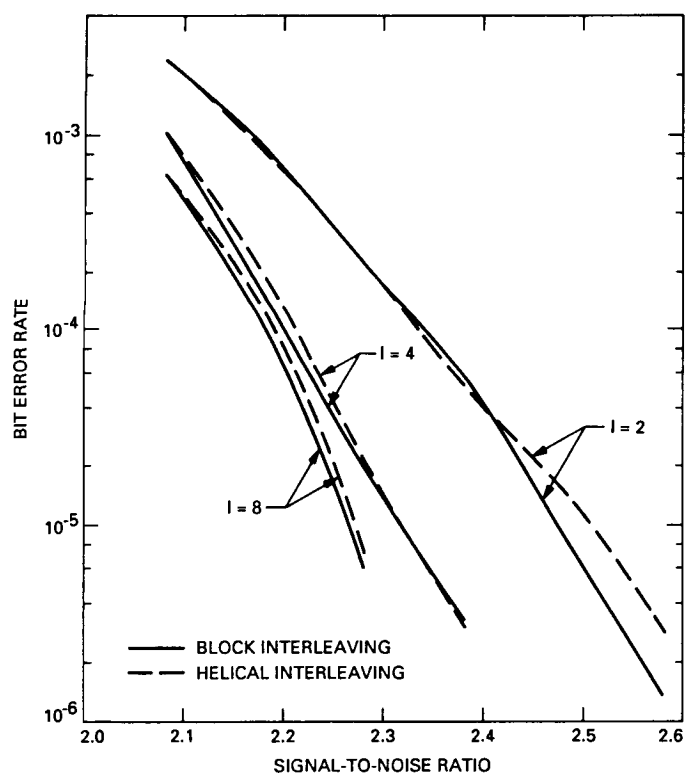


Fig. 4. (7,1/2) convolutional code and (255,223) Reed-Solomon code; no error forecasting.

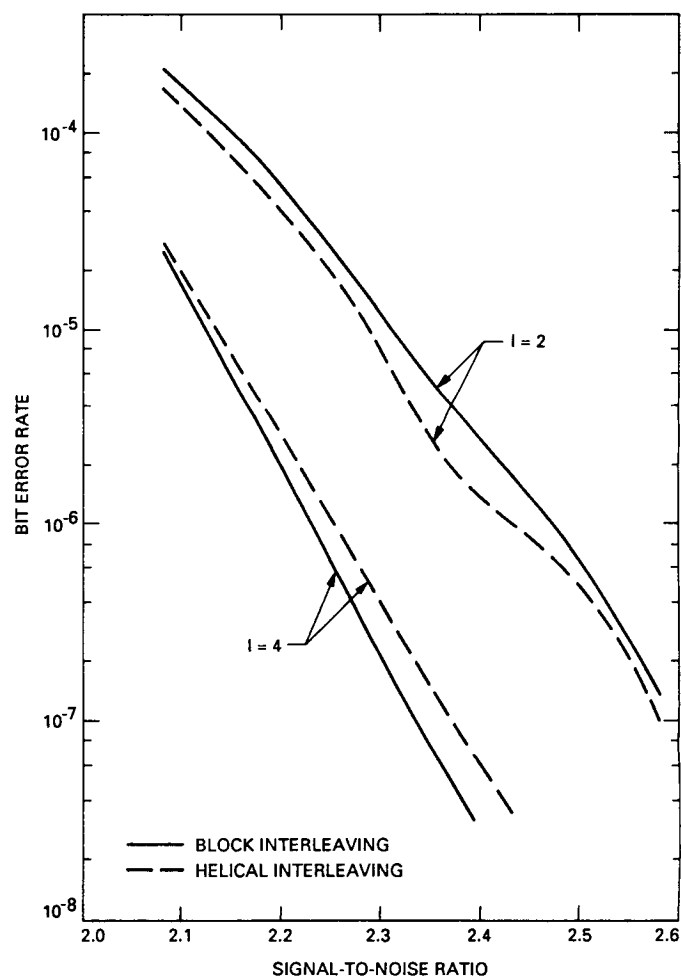


Fig. 5. (7,1/2) convolutional code and (255,223) Reed-Solomon code with error forecasting.

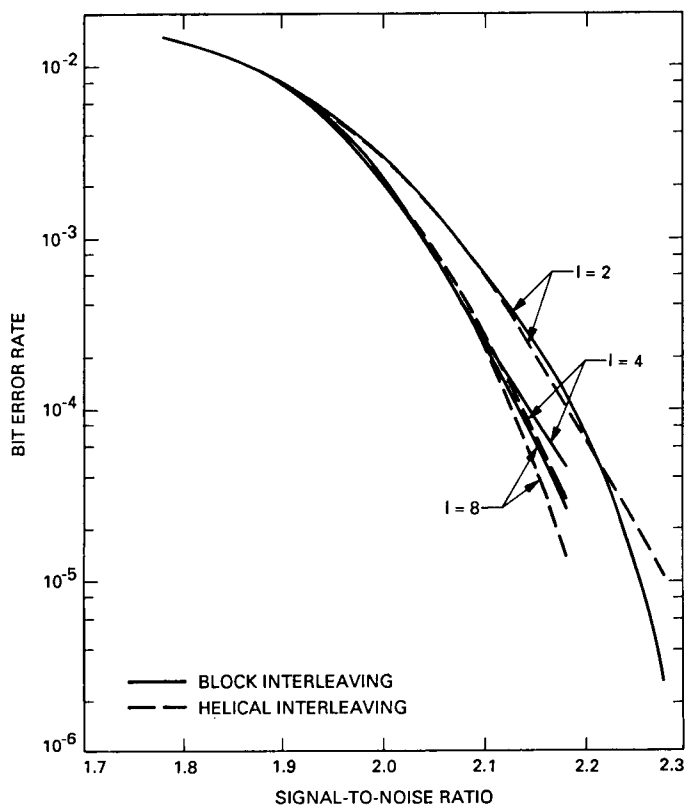


Fig. 6. (7,1/2) convolutional code and (1023,959) Reed-Solomon code; no error forecasting.

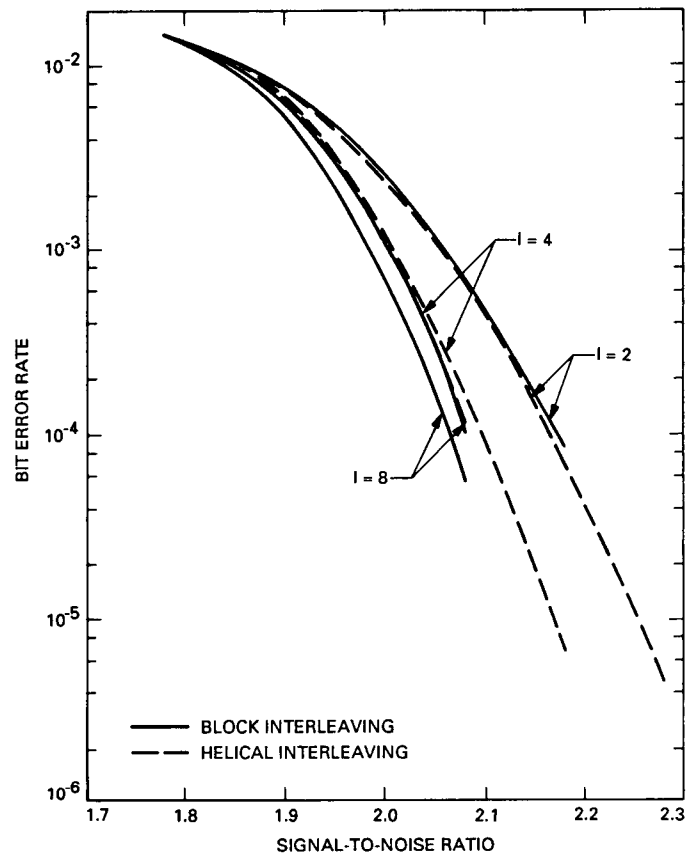


Fig. 7. (7,1/2) convolutional code and (1023,959) Reed-Solomon code with error forecasting.

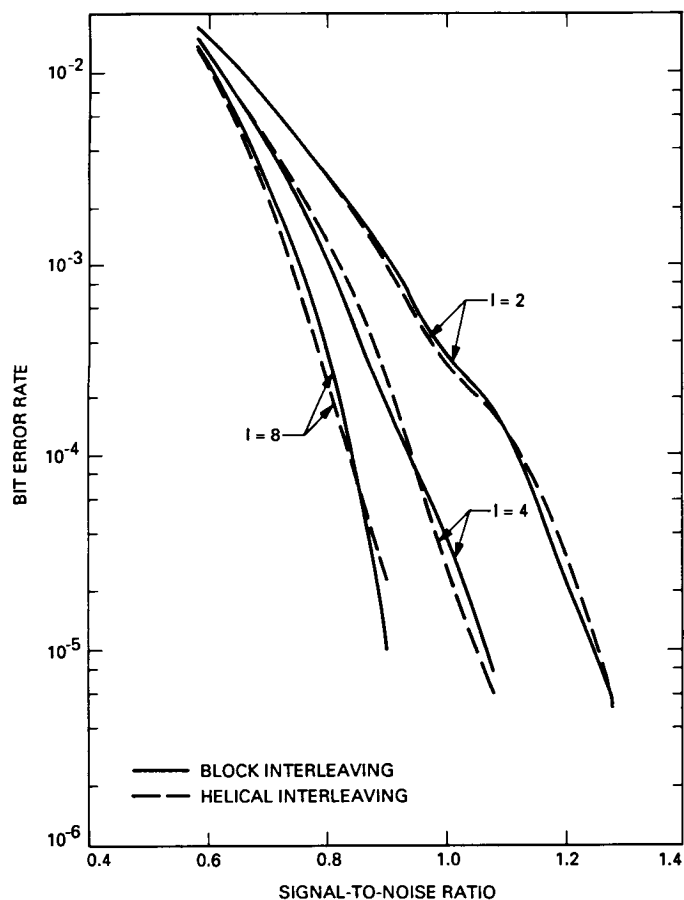


Fig. 8. (15,1/4) convolutional code and (255,223) Reed-Solomon code; no error forecasting.

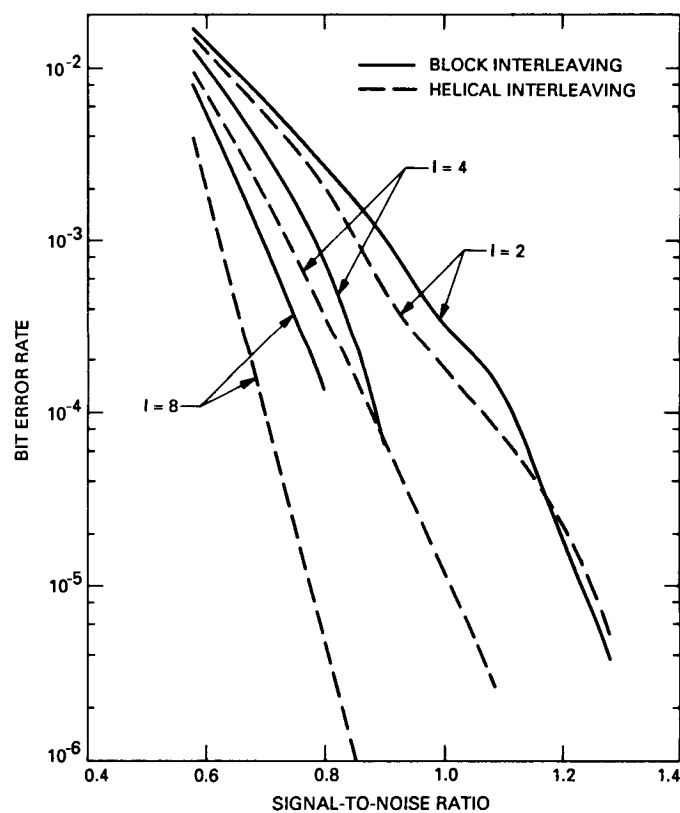
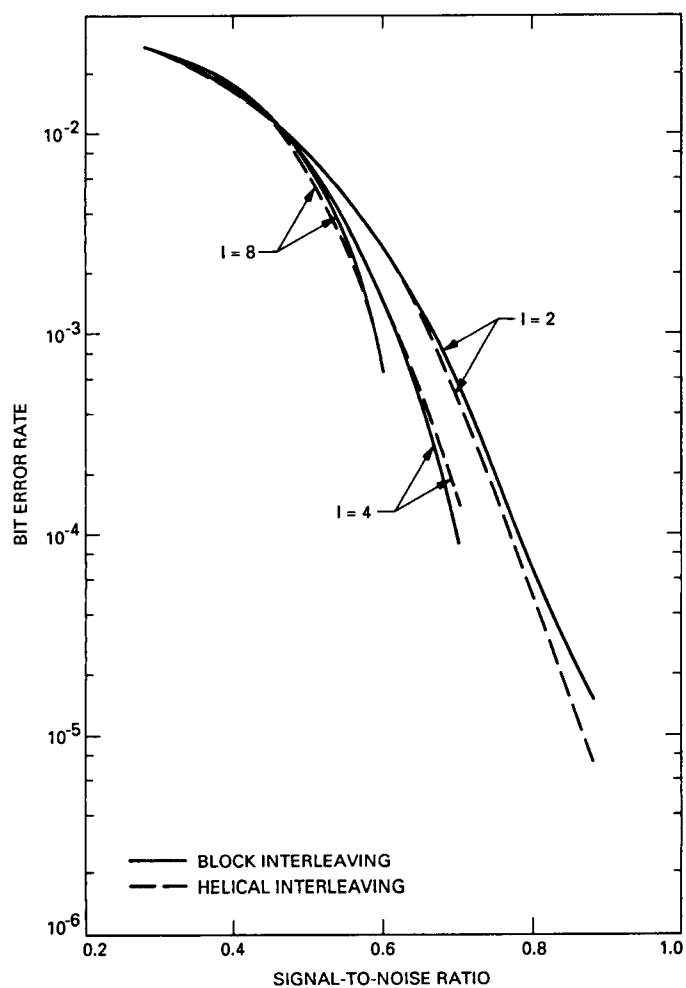
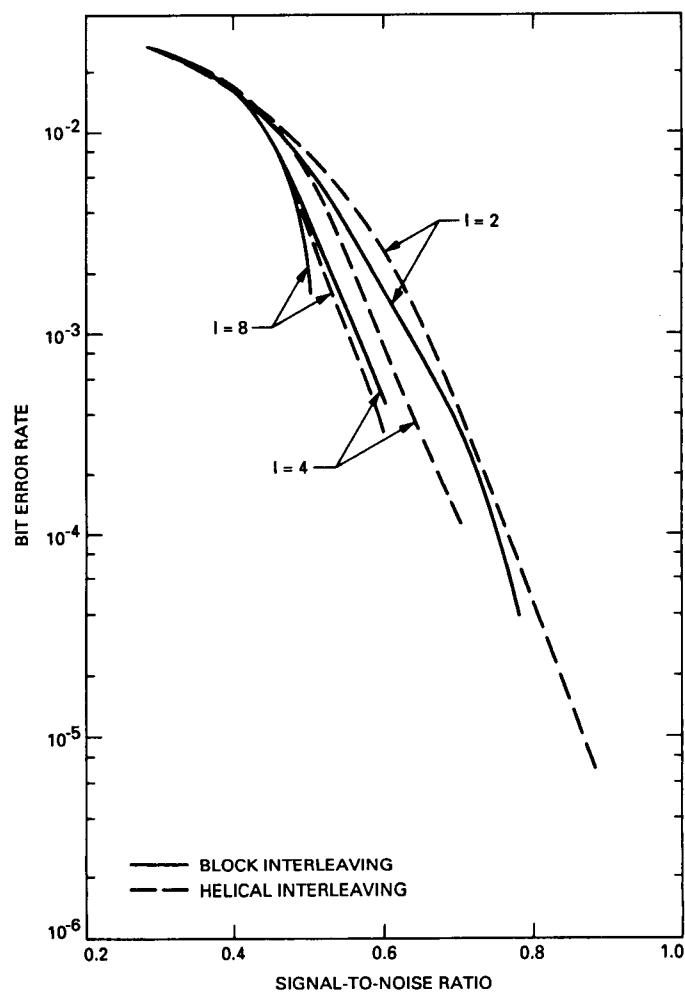


Fig. 9. (15,1/4) convolutional code (255,223) Reed-Solomon code with error forecasting.



**Fig. 10. (15,1/4) convolutional code (1023,959)
Reed-Solomon code; no error forecasting.**



**Fig. 11. (15,1/4) convolutional code (1023,959)
Reed-Solomon code with error forecasting.**